FatalErrors

Home Blogs Post a blog

Home > Detail page

## K3s offline installation - four types

preface
K3s is a lightweight Kubernetes distribution. In the statistics of k3s downloads in 2020, k3s has been downloaded more than 1 million times worldwide and installed more than 20000 times a week, of which 30% of the downloads come from China. In China, many users have applied k3s to various edge computing and Internet of things devices. At the same time, it is also widely used in production line robots deployed in intelligent factories and some of the world's largest wind power plants.

For K3s in production environment, an insurmountable problem is offline installation. In your offline environment, you need to prepare the following three components:

K3s installation script
K3s binary
K3s dependent image

The above three components can be accessed through the K3s Release page( https://github.com/k3s-io/k3s/releases )Download. If it is used in China, it is recommended to download it from http://mirror.cnrancher.com Get these components.
The author believes that the focus of offline installation lies in the image part that K3s depends on, because the "installation script" and "binary" of K3s only need to be downloaded to the corresponding directory and then given the corresponding permissions, which is very simple. However, the installation method of the image that K3s depends on depends on whether you use manual deployment image or private image warehouse, and also depends on whether container or docker is used when the container runs.

For different combination forms, it can be divided into the following forms to realize offline installation:

Container + manual deployment image mode
Docker + manual deployment image mode
Container + private image warehouse mode
Docker + private image warehouse

Container + manual deployment image mode
Suppose you have downloaded the K3s installation script (K3s install. SH), K3s binary (k3s) and K3s dependent image (k3s-airgap-images-amd64.tar) of the same version to the / root directory.

If the container you use is containerd at runtime, when you start K3s, it will check whether / var / lib / Ranger / K3s / agent / images / has an available image package. If so, import the image into the containerd image list. Therefore, we only need to download the K3s dependent image to / var / lib / Ranger / K3s / agent / images / directory, and then start K3s.

1. Import image to containerd image list
sudo mkdir -p /var/lib/rancher/k3s/agent/images/
sudo cp /root/k3s-airgap-images-amd64.tar /var/lib/rancher/k3s/agent/images/

2. Move the K3s installation script and K3s binary files to the corresponding directory and grant executable permissions
sudo chmod a+x /root/k3s /root/k3s-install.sh
sudo cp /root/k3s /usr/local/bin/

3. Install K3s
INSTALL_K3S_SKIP_DOWNLOAD=true /root/k3s-install.sh

After a moment, you can see that K3s has been started successfully:

```
root@k3s-docker:~# crictl images
IMAGE                                      TAG                 IMAGE ID            SIZE
docker.io/rancher/coredns-coredns          1.8.0               296a6d5035e2d       42.6MB
docker.io/rancher/klipper-helm             v0.3.2              4be09ab862d40       146MB
docker.io/rancher/klipper-lb               v0.1.2              897ce3c5fc8ff       6.46MB
docker.io/rancher/library-busybox          1.31.1              1c35c44120825       1.44MB
docker.io/rancher/library-traefik          1.7.19              aa764f7db3051       86.6MB
docker.io/rancher/local-path-provisioner   v0.0.14             e422121c9c5f9       42MB
docker.io/rancher/metrics-server           v0.3.6              9dd718864ce61       41.2MB
docker.io/rancher/pause                    3.1                 da86e6ba6ca19       746kB

root@k3s-docker:~# kubectl get pods -A
NAMESPACE     NAME                                      READY   STATUS      RESTARTS   AGE
kube-system   local-path-provisioner-7c458769fb-zdg9z   1/1     Running     0          38s
kube-system   coredns-854c77959c-696gk                  1/1     Running     0          38s
kube-system   metrics-server-86cbb8457f-hs6vw           1/1     Running     0          38s
kube-system   helm-install-traefik-4pgcr                0/1     Completed   0          38s
kube-system   svclb-traefik-bq7wl                       2/2     Running     0          17s
kube-system   traefik-6f9cbd9bd4-jccd7                  1/1     Running     0          17s
```

Docker + manual deployment image mode
Suppose you have downloaded the K3s installation script (K3s install. SH), K3s binary (k3s) and K3s dependent image

Hot Categories

Java × 321
Android × 221
Linux × 182
Python × 111
MySQL × 103
Programming × 101
Javascript × 98
Database × 52
Big Data × 45
Oracle × 45
iOS × 43

Hot Tags

Java × 8678
Python × 3398
Algorithm × 2157
Linux × 2069
Javascript × 1932
data structure × 1524
Spring × 1497
C++ × 1439
MySQL × 1163
Database × 1138
Front-end × 1057
Design Pattern × 1024

Suppose you have downloaded the K3s installation script (K3s install. SH), K3s binary (k3s) and K3s dependent image (k3s-airgap-images-amd64.tar) of the same version to the / root directory.

Unlike containerd, when docker is used as a container to run, K3s will not import the image in / var / lib / Ranger / K3s / agent / images / directory when you start K3s. Therefore, before starting K3s, we need to manually import the K3s dependent images into the docker image list.

1. Import image to docker image list
sudo docker load -i /root/k3s-airgap-images-amd64.tar

2. Move the K3s installation script and K3s binary files to the corresponding directory and grant executable permissions
sudo chmod a+x /root/k3s /root/k3s-install.sh
sudo cp /root/k3s /usr/local/bin/

3. Install K3s
INSTALL_K3S_SKIP_DOWNLOAD=true INSTALL_K3S_EXEC='--docker' /root/k3s-install.sh

After a moment, you can see that K3s has been started successfully:

```
root@k3s-docker:~# docker images
REPOSITORY                       TAG                 IMAGE ID            CREATED
rancher/klipper-helm             v0.3.2              4be09ab862d4        7 weeks ago
rancher/coredns-coredns          1.8.0               296a6d5035e2        2 months ago
rancher/library-busybox          1.31.1              1c35c4412082        7 months ago
rancher/local-path-provisioner   v0.0.14             e422121c9c5f        7 months ago
rancher/library-traefik          1.7.19              aa764f7db305        14 months ago
rancher/metrics-server           v0.3.6              9dd718864ce6        14 months ago
rancher/klipper-lb               v0.1.2              897ce3c5fc8f        19 months ago
rancher/pause                    3.1                 da86e6ba6ca1        3 years ago

root@k3s-docker:~# kubectl get pods -A
NAMESPACE     NAME                                      READY   STATUS      RESTARTS   AGE
kube-system   metrics-server-86cbb8457f-8ckr6           1/1     Running     0          30s
kube-system   local-path-provisioner-7c458769fb-vhkjr   1/1     Running     0          30s
kube-system   helm-install-traefik-4b46c                0/1     Completed   0          31s
kube-system   coredns-854c77959c-4ql8t                  1/1     Running     0          30s
kube-system   svclb-traefik-kbtbx                       2/2     Running     0          27s
kube-system   traefik-6f9cbd9bd4-rbm6k                  1/1     Running     0          27s
```

Container + private image warehouse mode
Suppose you have downloaded the K3s installation script (K3s install. SH) and K3s binary (k3s) of the same version to the / root directory. And the images required by K3s have been uploaded to the image warehouse (the address of the image warehouse in this example is: http://192.168.64.44:5000 ). The list of images required for K3s can be obtained from k3s-images.txt on the K3s Release page.

1. Configure the K3s mirror warehouse

When you start K3s, the image will be pulled from docker.io by default. When using containerd container to run and install offline, we only need to configure the image warehouse address to the endpoint under docker.io. For more configuration instructions, please refer to the complete strategy for configuring containerd image warehouse or the K3s official document:

```
https://docs.rancher.cn/docs/k ... ndex/
sudo mkdir -p /etc/rancher/k3s
sudo cat >> /etc/rancher/k3s/registries.yaml <<EOF
mirrors:
"docker.io":
endpoint:
  - "http://192.168.64.44:5000"
  - "https://registry-1.docker.io"
EOF
```

2. Move the K3s installation script and K3s binary files to the corresponding directory and grant executable permissions

```
sudo chmod a+x /root/k3s /root/k3s-install.sh
sudo cp /root/k3s /usr/local/bin/
```

3. Install K3s

```
INSTALL_K3S_SKIP_DOWNLOAD=true /root/k3s-install.sh
```

After a moment, you can see that K3s has been started successfully:

```
root@k3s-containerd:~# crictl images
IMAGE                                      TAG                 IMAGE ID            SIZE
docker.io/rancher/coredns-coredns          1.8.0               296a6d5035e2d       12.9MB
docker.io/rancher/klipper-helm             v0.3.2              4be09ab862d40       50.7MB
docker.io/rancher/klipper-lb               v0.1.2              897ce3c5fc8ff       2.71MB
docker.io/rancher/library-traefik          1.7.19              aa764f7db3051       24MB
docker.io/rancher/local-path-provisioner   v0.0.14             e422121c9c5f9       13.4MB
docker.io/rancher/metrics-server           v0.3.6              9dd718864ce61       10.5MB
docker.io/rancher/pause                    3.1                 da86e6ba6ca19       326kB

root@k3s-containerd:~# kubectl get pods -A
NAMESPACE     NAME                                      READY   STATUS      RESTARTS   AGE
kube-system   local-path-provisioner-7c458769fb-7w8hb   1/1     Running     0          37s
kube-system   coredns-854c77959c-f8m2n                  1/1     Running     0          37s
kube-system   helm-install-traefik-9lbrx                0/1     Completed   0          38s
kube-system   svclb-traefik-x8f6f                       2/2     Running     0          29s
kube-system   metrics-server-86cbb8457f-f7lb7           1/1     Running     0          37s
kube-system   traefik-6f9cbd9bd4-4s66r                  1/1     Running     0          29s
```

Docker + private image warehouse

Suppose you have downloaded the K3s installation script (K3s install. SH) and K3s binary (k3s) of the same version to the / root directory. And the images required by K3s have been uploaded to the image warehouse (the address of the image warehouse in this example is: http://192.168.64.44:5000 ). The list of images required for K3s can be obtained from k3s-images.txt on the K3s Release page.

1. Configure the K3s mirror warehouse

Docker does not support containerd. You can modify the corresponding endpoint of docker.io (the default is https://registry-1.docker.io )To indirectly modify the address of the default image warehouse. However, in docker, you can configure registry mirrors to obtain K3s images from other image warehouses. After this configuration, we will first pull the image from the address configured by registry mirrors. If we can't get it, we will get the image from the default docker.io, so as to meet our needs.

```
cat >> /etc/docker/daemon.json <<EOF
{
"registry-mirrors": ["http://192.168.64.44:5000"]
}
EOF

sudo systemctl daemon-reload
sudo systemctl restart docker
```

2. Move the K3s installation script and K3s binary files to the corresponding directory and grant executable permissions

```
sudo chmod a+x /root/k3s /root/k3s-install.sh
sudo cp /root/k3s /usr/local/bin/
```

3. Install K3s

```
INSTALL_K3S_SKIP_DOWNLOAD=true INSTALL_K3S_EXEC='--docker' /root/k3s-install.sh
```

After a moment, you can see that K3s has been started successfully:

```
root@k3s-docker:~# docker images
REPOSITORY                        TAG                 IMAGE ID            CREATED
rancher/klipper-helm              v0.3.2              4be09ab862d4        7 weeks ago
rancher/coredns-coredns           1.8.0               296a6d5035e2        2 months ago
rancher/local-path-provisioner    v0.0.14             e422121c9c5f        7 months ago
rancher/library-traefik           1.7.19              aa764f7db305        14 months ago
rancher/metrics-server            v0.3.6              9dd718864ce6        14 months ago
rancher/klipper-lb                v0.1.2              897ce3c5fc8f        19 months ago
rancher/pause                     3.1                 da86e6ba6ca1        3 years ago

root@k3s-docker:~# kubectl get pods -A
NAMESPACE     NAME                                      READY   STATUS      RESTARTS   AGE
kube-system   helm-install-traefik-bcclh                0/1     Completed   0          33s
kube-system   coredns-854c77959c-kp85f                  1/1     Running     0          33s
kube-system   metrics-server-86cbb8457f-85fpd           1/1     Running     0          33s
kube-system   local-path-provisioner-7c458769fb-r5nkw   1/1     Running     0          33s
kube-system   svclb-traefik-rbmhk                       2/2     Running     0          24s
kube-system   traefik-6f9cbd9bd4-k6t9n                  1/1     Running     0          24s
```

Postscript
The manual deployment image method is more suitable for small-scale installation and a small number of nodes. Private image warehouse is more suitable for clusters with large scale and many nodes. The docker registry in this article is built in the simplest way. Docker run - D - P 5000:5000 -- restart = always -- name registry: 2. You may need to modify some parameters about registry due to different building methods of image warehouse in your environment.

Tag: Docker Kubernetes Container k3s Posted by **zhushuyun** at Nov 02, 2021 - 3:16 AM